Re

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/782,672	02/19/2004	Gordon D. McIntosh	AUS920031082US1	5246

35525 7590 12/31/2007

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
HA, LEYNNA A	
ART UNIT	PAPER NUMBER
2135	
MAIL DATE	DELIVERY MODE
12/31/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/782,672	**Applicant(s)** MCINTOSH, GORDON D.	
	Examiner LEYNNA T. HA	**Art Unit** 2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 October 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-25 are pending.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1-24 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1 and 21 recites a method in a data processing system for processing instructions by a processing unit that has or use a standard instruction set and processing only those instructions that use the new instruction set. The claimed suggest using a software program for processing instructions that use the new instruction set. Therefore, claims 1 and 21 are directed to a program per se.

Claim 11 recites a computer program product, which is stored in a computer recordable medium. Although, claim 11 recites a computer recordable medium, this medium can also be a computer readable media that is directed to non-functional descriptive material used in a data processing system. Specification discloses on pg.28, the computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system. Thus, examiner gives the broadest interpretation for the claimed medium is the form of coded formats. Therefore, claim 11 is directed to a program per se.

MPEP: **2106.01 [R-5]** **> Computer-Related Nonstatutory Subject Matter<

>Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and ***computer programs* which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works, and a compilation or mere arrangement of data.

Response to Arguments

3. Applicant's arguments with respect to claims 1-25 have been considered but are moot in view of the new ground(s) of rejection.

Claims 1-25 is currently rejected over Zaidi, et al. (US 6,542,981) in view of Pechanek, et al. (US 6,848,041). Pechanek remains the secondary prior art disclosing the obviousness to dynamically remap the instruction set. It would have been obvious for a person of ordinary skills in the art to combine the teaching of Zaidi with Pechanek teaching to dynamically remap the instruction set because the ability to dynamically create a set of instructions on a task by task basis is for the primary purpose of improving control and parallel code density (Pechanek – col. 12, lines 6-29) which contain new instructions that provide optimized higher performance, improved code density, and new functionality (Pechanek – col.1, lines 44-61).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. **Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zaidi, et al. (US 6,542,981), and further in view of Pechanek, et al. (US 6,848,041).**

As per claim 1:

Zaidi, et al. discusses a method in a data processing system for processing instructions by a processing unit, the method comprising:

A method in a data processing system for processing instructions by a processing unit that has a standard instruction set, the method comprising:

using, by an encryption algorithm (col.7, lines 25-60 and col.9, lines 1-7) each time the data processing system is rebooted (col.3, lines 15-20 and col.5, lines 28-30 and 46-60), a different one of a plurality of different instruction maps to *[dynamically]* remap the standard instruction set (col.6, lines 8-18; instructions prior to replacing with new instructions or traditional instructions are given as standard instruction set) to create a new instruction set; and (col.4, lines 44-47 and col.5, lines 14-17)

processing, by the processing unit, only those instructions that use the new instruction set. (col.5, lines 19-23 and col.8, lines 20-27)

Zaidi teaches an invention that adds functionality to the typical BIOS start sequence to provide microcode upgrade to the processor by adding a special RISC instruction and a set of microcode instructions to the chip containing the BIOS (col.3, lines 15-20). This suggests the invention is in the time the data process system is rebooted. Zaidi suggests remapping to create new instruction set where the special function is a processor upgrade to upgrade the microcode to enhance the functionality of the process to replace the entire set of microcode instructions, to add a secure microcode instructions, to add compatibility with another instruction set, or to access hardware features (col.4, lines 27-32 and 44-48 and col.5, lines 14-17). Zaidi discloses the new BIOS instructions include transfer instructions and causes the set of microcode instructions to be written to the firmware suggesting the processing the instructions that use the new instruction set (col.5, lines 19-23 and col.8, lines 20-27). In addition, Zaidi discusses the upgrade will be detected by the already existing (standard) instruction in the BIOS during booting, restart and upon every boot up the set of microcode instructions is transferred (col.5, lines 28-30 and 46-60). Thereafter, the execution engine begins by executing instructions in the BIOS (col.6, lines 40-45), which is the resident microcode that causes execution to begin with authentication and decryption of the set of microcode instructions (col.7, lines 12-30 and col.9, lines 1-7). The BIOS contains the digital certificate and digital signature which are processed by the encryption engine in conjunction with the microcode upgrade or the set of microcode instructions (col.7, lines 30-60). Hence, Zaidi reads on the claimed using, by an encryption algorithm each time the data processing is rebooted, a different one of a plurality of different instruction maps to remap the standard instruction set and processing only those instructions that use the new instruction set. Although, Zaidi discloses remapping the standard instruction set to create a new instruction set but vaguely suggest dynamically remap such that the set of instructions would be executed securely, privately, and without interruption (col.2, lines 58-62). Thus, Zaidi did not fully disclose to dynamically remap the instruction set.

Pechanek, et al. discloses an invention that solves the problem of instruction set scability by defining a hierarchical instruction set that allows application specific processors to be developed which contain new instructions that provide optimized capabilities for specific applications (col.1, lines 44-50). These capabilities can result in higher performance, improved code density, and new functionality and pluggable relates to groups of instructions that can easily be added to a processor architecture for code density and performance enhancements (col.1, lines 50-55). Pechanek includes compacted instruction set which allows the ability to dynamically create a set of compacted instructions on a task by task basis for the primary purpose of improving control and parallel density (col.1, lines 58-67 and col.12, lines 15-18). Hence, Pechanek suggests to dynamically remap the standard instruction set to create a new instruction set. Pechanek discloses specific bit encoding within Instruction Type (col.6, lines 8-10), simple fixed translations to a known state for a given instruction mapping (col.10, lines 60-61 and col.13, lines 50-67), and the ability to independently select instructions within 3-bit address range in each function VIM (col.14, lines 30-38). Additionally, Pechanek discloses the use of an enhanced tool to support application analysis and the instruction selection process, is deemed advantageous (col.12, lines 26-29).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of Zaidi with Pechanek teaching to dynamically remap the instruction set because the ability to dynamically create a set of instructions on a task by task basis is for the primary purpose of improving control and parallel code density (Pechanek – col. 12, lines 6-29) which contain new instructions that provide optimized higher performance, improved code density, and new functionality (Pechanek – col.1, lines 44-61).

As per claim 2: See Zaidi on col.2,lines 58-62 and Pechanek on col.1, lines 49-50; discussing the method of claim 1, further comprising: performing the dynamic remapping during execution of an initial program load (IPL) process and before the data processing system begins executing an operating system.

As per claim 3: See Zaidi on col.1, lines 28-38; discussing the method of claim 1, wherein each one of the plurality of different instruction maps is an opcode map.

As per claim 4: See Zaidi on col.7, lines 25-60 and col.9, lines 1-7; discussing the method of claim 1 further comprising: encoding a set of instructions from a trusted computer base using the one of the plurality of different instruction maps form a set of encoded instructions; and sending the set of encoded instructions to the processing unit for execution.

As per claim 5: See Zaidi on col.2,lines 42-46; discussing the method of claim 1, wherein the processing unit is at least one processor.

As per claim 6: See Zaidi on col.7, lines 25-60 and col.9, lines 1-7; discussing the method of claim 4, wherein the encoding step and the sending step are performed by a program loader.

As per claim 7: See Zaidi on col.5, lines 58-65 and col.8, lines 40-65; discussing the method of claim 1 further comprising: responsive to an event, executing a process to select the one of the plurality of different instruction maps selected instruction map.

As per claim 8: See Zaidi on col.3, lines 1-24 and col.5, lines 58-65 and Pechanek on col.15,lines 20-55; discussing the method of claim 7, wherein the process uses a machine serial number and a number of boot cycles to select the one of the plurality of different instruction maps.

As per claim 9: See Zaidi on col.3, lines 10-35 and col.5, lines 9-10; discussing the method of claim 7, wherein the event is at least one of an initialization of the data processing system and a user input.

As per claim 10: See Zaidi on col.7,lines 30-60 and col.8,lines 40-65; discussing the method of claim 1, wherein the new instruction set is created using a first one of the plurality of different instruction maps when code is executed by a first privilege level and wherein a second one of the plurality of different instruction maps is used when code is executed by a second privilege level.

As per claim 11:

Zaidi, et al. discusses a computer program product, which is stored in a computer recordable medium for processing instructions by a processing unit, which has a standard instruction set, in a data processing system, the computer program product comprising:

first instructions for using, by an encryption algorithm (col.7, lines 25-60 and col.9, lines 1-7) each time the data processing system is rebooted (col.3, lines 15-20 and col.5, lines 28-30 and 46-60), a different one of a plurality of different instruction maps to *[dynamically]* remap the standard instruction set (col.6, lines 8-18; instructions prior to replacing with new instructions or traditional instructions are given as standard instruction set) to create a new instruction set; and(col.4, lines 44-47 and col.5, lines 14-17)

second instructions for processing, by the processing unit, only those instructions that use the new instruction set. (col.5, lines 19-23 and col.8, lines 20-27)

Zaidi teaches an invention that adds functionality to the typical BIOS start sequence to provide microcode upgrade to the processor by adding a special RISC instruction and a set of microcode instructions to the chip containing the BIOS (col.3, lines 15-20). This suggests the invention is in the time the data process system is rebooted. Zaidi suggests remapping to create new instruction set where the special function is a processor upgrade to upgrade the microcode to enhance the functionality of the process to replace the entire set of microcode instructions, to add a secure microcode instructions, to add compatibility with another instruction set, or to access hardware features (col.4, lines 27-32 and 44-48 and col.5, lines 14-17). Zaidi discloses the new BIOS instructions include transfer instructions and causes the set of microcode instructions to be written to the firmware suggesting the processing the instructions that use the new instruction set (col.5, lines 19-23 and col.8, lines 20-27). In addition, Zaidi discusses the upgrade will be detected by the already existing (standard) instruction in the BIOS during booting, restart and upon every boot up the set of microcode instructions is transferred (col.5, lines 28-30 and 46-60). Thereafter, the execution engine begins by executing instructions in the BIOS (col.6, lines 40-45), which is the resident microcode that causes execution to begin with authentication and decryption of the set of microcode instructions (col.7, lines 12-30 and col.9, lines 1-7). The BIOS contains the digital certificate and digital signature which are processed by the encryption engine in conjunction with the microcode upgrade or the set of microcode instructions (col.7, lines 30-60). Hence, Zaidi reads on the claimed using, by an encryption algorithm each time the data processing is rebooted, a different one of a plurality of different instruction maps to remap the standard instruction set and processing only those instructions that use the new instruction set. Although, Zaidi discloses remapping the standard instruction set to create a new instruction set but vaguely suggest dynamically remap such that the set of instructions would be executed securely, privately, and without interruption (col.2, lines 58-62). Thus, Zaidi did not fully disclose to dynamically remap the instruction set.

Pechanek, et al. discloses an invention that solves the problem of instruction set scability by defining a hierarchical instruction set that allows application specific processors to be developed which contain new instructions that provide optimized capabilities for specific applications (col.1, lines 44-50). These capabilities can result in higher performance, improved code density, and new functionality and pluggable relates to groups of instructions that can easily be added to a processor architecture for code density and performance enhancements (col.1, lines 50-55). Pechanek includes compacted instruction set which allows the ability to dynamically create a set of compacted instructions on a task by task basis for the primary purpose of improving control and parallel density (col.1, lines 58-67 and col.12, lines 15-18). Hence, Pechanek suggests to dynamically remap the standard instruction set to create a new instruction set. Pechanek discloses specific bit encoding within Instruction Type (col.6, lines 8-10), simple fixed translations to a known state for a given instruction mapping (col.10, lines 60-61 and col.13, lines 50-67), and the ability to independently select instructions within 3-bit address range in each function VIM (col.14, lines 30-38). Additionally, Pechanek discloses the use of an enhanced tool to support application analysis and the instruction selection process, is deemed advantageous (col.12, lines 26-29).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of Zaidi with Pechanek teaching to dynamically remap the instruction set because the ability to dynamically create a set of instructions on a task by task basis is for the primary purpose of improving control and parallel code density (Pechanek – col. 12, lines 6-29) which contain new instructions that provide optimized higher performance, improved code density, and new functionality (Pechanek – col.1, lines 44-61).

As per claim 12: See Zaidi on col.2,lines 58-62 and Pechanek on col.1, lines 49-50; discussing the computer program product of claim 11, further comprising: third instructions for performing the dynamic remapping during execution of an initial program load (IPL) process and before the data processing system begins executing an operating system.

As per claim 13: See Zaidi on col.1, lines 28-38; discussing the computer program product of claim 11, wherein each one of the plurality of different instruction maps is an opcode map.

As per claim 14: See Zaidi on col.7, lines 25-60 and col.9, lines 1-7; discussing the computer program product of claim 11 further comprising: third instructions for encoding a set of instructions from a trusted computer base using the one of the plurality of different instruction maps to form a set of encoded instructions; and fourth instructions for sending the set of encoded instructions to the processing unit for execution.

As per claim 15: See Zaidi on col.2,lines 42-46; discussing the computer program product of claim 11, wherein the processing unit is at least one processor.

As per claim 16: See Zaidi on col.7, lines 25-60 and col.9, lines 1-7; discussing the computer program product of claim 14, wherein the third instructions and the fourth instructions are performed by a program loader.

As per claim 17: See Zaidi on col.5, lines 58-65 and col.8, lines 40-65; discussing the computer program product of claim 11 further comprising: third instructions, responsive to an event, for executing a process to select the one of the plurality of different instruction maps.

As per claim 18: See Zaidi on col.3, lines 1-24 and col.5, lines 58-65 and Pechanek on col.15,lines 20-55; discussing the computer program product of claim 17, wherein the process uses a machine serial number and a number of boot cycles to select the one of the plurality of different instruction maps.

As per claim 19: See Zaidi on col.3, lines 10-35 and col.5, lines 9-10; discussing the computer program product of claim 17, wherein the event is at least one of an initialization of the data processing system and a user input.

As per claim 20: See Zaidi on col.7,lines 30-60 and col.8,lines 40-65; discussing the computer program product of claim 11, wherein the new instruction set is created using a first one of the plurality of different instruction maps when code is executed by a first privilege level and wherein a second one of the plurality of different instruction maps is used when code is executed by a second privilege level.

As per claim 21:

Zaidi, et al. discusses a data processing system for processing instructions by a processing unit that uses a standard instruction set, the data processing system comprising:

remapping means for using, by an encryption algorithm (col.7, lines 25-60 and col.9, lines 1-7) each time the data processing system is rebooted (col.3, lines 15-20 and col.5, lines 28-30 and 46-60), a different one of a plurality of different instruction maps to *[dynamically]* remap the standard instruction set (col.6, lines 8-18; instructions prior to replacing with new instructions or traditional instructions are given as standard instruction set) to create a new instruction set; and (col.4, lines 44-47 and col.5, lines 14-17) processing means for processing, by the processing unit, only those instructions that use the new instruction set. (col.5, lines 19-23 and col.8, lines 20-27)

Zaidi teaches an invention that adds functionality to the typical BIOS start sequence to provide microcode upgrade to the processor by adding a special RISC instruction and a set of microcode instructions to the chip containing the BIOS (col.3, lines 15-20). This suggests the invention is in the time the data process system is rebooted. Zaidi suggests remapping to create new instruction set where the special function is a processor upgrade to upgrade the microcode to enhance the functionality of the process to replace the entire set of microcode instructions, to add a secure microcode instructions, to add compatibility with another instruction set, or to access hardware features (col.4, lines 27-32 and 44-48 and col.5, lines 14-17). Zaidi discloses the new BIOS instructions include transfer instructions and causes the set of microcode instructions to be written to the firmware suggesting the processing the instructions that use the new instruction set (col.5, lines 19-23 and col.8, lines 20-27). In addition, Zaidi discusses the upgrade will be detected by the already existing (standard) instruction in the BIOS during booting, restart and upon every boot up the set of microcode instructions is transferred (col.5, lines 28-30 and 46-60). Thereafter, the execution engine begins by executing instructions in the BIOS (col.6, lines 40-45), which is the resident microcode that causes execution to begin with authentication and decryption of the set of microcode instructions (col.7, lines 12-30 and col.9, lines 1-7). The BIOS contains the digital certificate and digital signature which are processed by the encryption engine in conjunction with the microcode upgrade or the set of microcode instructions (col.7, lines 30-60). Hence, Zaidi reads on the claimed using, by an encryption algorithm each time the data processing is rebooted, a different one of a plurality of different instruction maps to remap the standard instruction set and processing only those instructions that use the new instruction set. Although, Zaidi discloses remapping the standard instruction set to create a new instruction set but vaguely suggest dynamically remap such that the set of instructions would be executed securely, privately, and without interruption (col.2, lines 58-62). Thus, Zaidi did not fully disclose to dynamically remap the instruction set.

Pechanek, et al. discloses an invention that solves the problem of instruction set scability by defining a hierarchical instruction set that allows application specific processors to be developed which contain new instructions that provide optimized capabilities for specific applications (col.1, lines 44-50). These capabilities can result in higher performance, improved code density, and new functionality and pluggable relates to groups of instructions that can easily be added to a processor architecture for code density and performance enhancements (col.1, lines 50-55). Pechanek includes compacted instruction set which allows the ability to dynamically create a set of compacted instructions on a task by task basis for the primary purpose of improving control and parallel density (col.1, lines 58-67 and col.12, lines 15-18). Hence, Pechanek suggests to dynamically remap the standard instruction set to create a new instruction set. Pechanek discloses specific bit encoding within Instruction Type (col.6, lines 8-10), simple fixed translations to a known state for a given instruction mapping (col.10, lines 60-61 and col.13, lines 50-67), and the ability to independently select instructions within 3-bit address range in each function VIM (col.14, lines 30-38). Additionally, Pechanek discloses the use of an enhanced tool to support application analysis and the instruction selection process, is deemed advantageous (col.12, lines 26-29).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of Zaidi with Pechanek teaching to dynamically remap the instruction set because the ability to dynamically create a set of instructions on a task by task basis is for the primary purpose of improving control and parallel code density (Pechanek – col. 12, lines 6-29) which contain new instructions that provide optimized higher performance, improved code density, and new functionality (Pechanek – col.1, lines 44-61).

As per claim 22: See Zaidi on col.7, lines 30-60 and col.8,lines 40-65; discussing the data processing system of claim 21, wherein a new instruction map is selected each time the data processing system is started.

As per claim 23: See Zaidi on col.1, lines 28-38; discussing the data processing system of claim 21, wherein each one of the plurality of different instruction maps is an opcode map.

As per claim 24: See Zaidi on col.7, lines 25-60 and col.9, lines 1-7; discussing the data processing system of claim 21 further comprising: encoding means for encoding a set of instructions from a trusted computer base using the one of the plurality of different instruction maps to form a set of encoded instructions; and sending means for sending the set of encoded instructions to the processing unit for execution.

As per claim 25:

Zaidi, et al. discusses a data processing system comprising:

a bus system; (col.3, lines 56-65)

a memory connected to the bus system, wherein the memory includes a set of instructions; and (col.3, lines 66-67)

a processing unit, which has a standard instruction set, connected to the bus system, wherein the processing unit executes a set of instructions to use, by an encryption algorithm (col.7, lines 25-60 and col.9, lines 1-7) each time the data processing system is rebooted (col.3, lines 15-20 and col.5, lines 28-30 and 46-60), a different one of a plurality of different instruction maps to *[dynamically]* remap the standard instruction set (col.6, lines 8-18; instructions prior to replacing with new instructions or traditional instructions are given as standard instruction set) to create a new instruction set; and (col.4, lines 44-47 and col.5, lines 14-17)

process only those instructions that use the new instruction set. (col.5, lines 19-23 and col.8, lines 20-27)

Zaidi teaches an invention that adds functionality to the typical BIOS start sequence to provide microcode upgrade to the processor by adding a special RISC instruction and a set of microcode instructions to the chip containing the BIOS (col.3, lines 15-20). This suggests the invention is in the time the data process system is rebooted. Zaidi suggests remapping to create new instruction set where the special function is a processor upgrade to upgrade the microcode to enhance the functionality of the process to replace the entire set of microcode instructions, to add a secure microcode instructions, to add compatibility with another instruction set, or to access hardware features (col.4, lines 27-32 and 44-48 and col.5, lines 14-17). Zaidi discloses the new BIOS instructions include transfer instructions and causes the set of microcode instructions to be written to the firmware suggesting the processing the instructions that use the new instruction set (col.5, lines 19-23 and col.8, lines 20-27). In addition, Zaidi discusses the upgrade will be detected by the already existing (standard) instruction in the BIOS during booting, restart and upon every boot up the set of microcode instructions is transferred (col.5, lines 28-30 and 46-60). Thereafter, the execution engine begins by executing instructions in the BIOS (col.6, lines 40-45), which is the resident microcode that causes execution to begin with authentication and decryption of the set of microcode instructions (col.7, lines 12-30 and col.9, lines 1-7). The BIOS contains the digital certificate and digital signature which are processed by the encryption engine in conjunction with the microcode upgrade or the set of microcode instructions (col.7, lines 30-60). Hence, Zaidi reads on the claimed using, by an encryption algorithm each time the data processing is rebooted, a different one of a plurality of different instruction maps to remap the standard instruction set and processing only those instructions that use the new instruction set. Although, Zaidi discloses remapping the standard instruction set to create a new instruction set but vaguely suggest dynamically remap such that the set of instructions would be executed securely, privately, and without interruption (col.2, lines 58-62). Thus, Zaidi did not fully disclose to dynamically remap the instruction set.

Pechanek, et al. discloses an invention that solves the problem of instruction set scability by defining a hierarchical instruction set that allows application specific processors to be developed which contain new instructions that provide optimized capabilities for specific applications (col.1, lines 44-50). These capabilities can result in higher performance, improved code density, and new functionality and pluggable relates to groups of instructions that can easily be added to a processor architecture for code density and performance enhancements (col.1, lines 50-55). Pechanek includes compacted instruction set which allows the ability to dynamically create a set of compacted instructions on a task by task basis for the primary purpose of improving control and parallel density (col.1, lines 58-67 and col.12, lines 15-18). Hence, Pechanek suggests to dynamically remap the standard instruction set to create a new instruction set. Pechanek discloses specific bit encoding within Instruction Type (col.6, lines 8-10), simple fixed translations to a known state for a given instruction mapping (col.10, lines 60-61 and col.13, lines 50-67), and the ability to independently select instructions within 3-bit address range in each function VIM (col.14, lines 30-38). Additionally, Pechanek discloses the use of an enhanced tool to support application analysis and the instruction selection process, is deemed advantageous (col.12, lines 26-29).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of Zaidi with Pechanek teaching to dynamically remap the instruction set because the ability to dynamically create a set of instructions on a task by task basis is for the primary purpose of improving control and parallel code density

(Pechanek – col. 12, lines 6-29) which contain new instructions that provide optimized higher performance, improved code density, and new functionality (Pechanek – col.1, lines 44-61).

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to LEYNNA T. HA whose telephone number is (571) 272-3851. The examiner can normally be reached on Monday - Thursday (7:00 - 5:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

LHa

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100